# Operating Systems Lab
# Assignment 6.

Assignment Given on: March 24, 2015.
Assignment Deadline: April 07, 2015

VIRTUAL MEMORY MANAGMENT

**Objective: Simulate and analyze the performance of page replacement techniques: LRU, clock, and FIFO** *by varying the page size*. **You have to use pure-demand paging technique with pre-paging feature.**

You have to simulate the functions of an MMU (Memory Management Unit).
Virtual memory management requires using a page table for logical to physical address translation. A typical page table would have the following structure.

| Page # | Frame # | Page Size | Valid/ Invalid | Reference Bit | Modify Bit | Clock/history bits |
|---|---|---|---|---|---|---|
| ... | ... | ... | | ... | ... | ... |
| ... | ... | ... | | ... | ... | ... |

1. You need to simulate the available main memory (contiguous physical memory as an array of bytes) as an array of bytes in your program. This array is of fixed size: say 512 bytes. The size of each frame can be 1 byte, 2 bytes, 4 bytes, and so on. The number of frames which can be accommodated on the main memory depends on the frame size. Larger the frame size, lesser will be the number of frames.
2. In a multiprogrammed environment there will be several processes which need to be loaded onto the main memory. Each process has its address space. Assume that we have a set of processes which declare the "limits" of their logical address spaces (i.e. No logical address generated will exceed the specified limit.). All these processes are submitted to the system at time t=0.
3. Assuming that there is no portion of main memory taken up by the kernel, you have the entire main memory (the array of 512 bytes) available to be given in units of frames to the various processes.
4. Though we are not going to actually execute any process, we can simulate process execution by generating a stream of memory references; after all the main memory is going to see nothing else but a stream of physical addresses.
5. We assume that we have access to the logical addresses generated by a multiprogrammed CPU while executing the same set of processes. We know the process ID (which generated the address) and the address itself. This is a "trace list of memory references" which will drive our simulation.
6. The trace list has a sequence of "process ID, memory references" (logical address).
7. The logical address will be resolved to a physical address (<frame number, offset>) by using the page table. Since you have to use **pure demand** paging, if the memory reference is within a page which has not yet been loaded into the memory, there will be a page fault.
8. The page fault will be serviced using a page replacement scheme (FIFO, LRU(using history bits), clock (using reference bit)). In your simulation you won't be doing an actual page replacement. You need to simulate page replacement by just updating the page table(s) of the process(es).
9. Examine the effects of the **local page replacement** and the global page replacement policy.
10. Examine the effect of **prepaging**. Prepaging means that whenever there is a page fault, you fetch in not just the required page, but also the next page.
11. You can use the sample simulation files (progAddrLimits.txt, memoryref.txt) to conduct your experiments. The first file gives the set of processes (process id, address limits). The second file gives

the trace of memory references.

12. You may use suitable command line arguments to provide the options for the program and the parameter values.

**In your report:**

Record the number of page faults occurring for each page size. Plot a graph (page size v/s number of page faults) for each page replacement scheme. You may consider page sizes of 1, 2, 4, 8, 16, 32 bytes for repeating your experiments. For a given experiment the page size would remain fixed. There will be three curves for the three page replacement schemes: FIFO, clock, LRU.

With the pre-paging feature turned on, again plot the graph (page size v/s number of page faults) for each page replacement scheme. The data which you use for plotting the graphs in your report will be verified during your demonstration. Also analyze your observations.